Discovering motion hierarchies via tree-structured coding of trajectories

Juan-Manuel Pérez-Rúa, Tomas Crivelli, Patrick Pérez, Patrick Bouthemy

Discovering motion hierarchies via tree-structured coding of trajectories

Juan-Manuel Pérez-Rúa[1]
juanmanuel.perezrua@technicolor.com
Tomas Crivelli[1]
http://www.technicolor.com/en/tomas-crivelli
Patrick Pérez[1]
http://www.technicolor.com/en/patrick-perez
Patrick Bouthemy[2]
patrick.bouthemy@inria.fr

[1] Technicolor R&I
Cesson Sévigné, France

[2] Inria
Centre Rennes
Bretagne Atlantique, France

Abstract

The dynamic content of physical scenes is largely compositional, that is, the movements of the objects and of their parts are hierarchically organised and relate through composition along this hierarchy. This structure also prevails in the apparent 2D motion that a video captures. Accessing this visual motion hierarchy is important to get a better understanding of dynamic scenes and is useful for video manipulation. We propose to capture it through learned, tree-structured sparse coding of point trajectories. We leverage this new representation within an unsupervised clustering scheme to partition hierarchically the trajectories into meaningful groups. We show through experiments on motion capture data that our model is able to extract moving segments along with their organisation. We also present competitive results on the task of segmenting objects in video sequences from trajectories.

1 Introduction

Early works in biological vision found that visual systems decompose objects into parts through the analysis of motion nesting [20]. Johansson showed in particular that removing the motion of the main body from the image reveals the distinctive motion of its parts. Along the same lines, Gershman *et al*. [15] have recently proposed a computational model that can decompose dynamic sensory data into a hierarchy of components. The hierarchical decomposition of visual motion information is thus clearly identified as a key step in complex biological vision systems. In this paper, we propose to investigate these ideas in the context of video analysis, where important tasks like motion understanding and video object segmentation could benefit from them.

The compositional organisation of visual motion stems first from the physics of observed objects: an object moves relative to its environment, its parts may be in motion relative to it, some of them may form articulated kinematic chains and dynamic deformations, if any, add another layer to the final 3D movement of object fragments. Projected in the image

Figure 1: Hierarchical organisation of visual motions in a natural scene.

plane of the camera, this organisation persists. In addition, the movement of the camera itself introduces another component that affects the apparent motion of all the visible parts of the scene. As a consequence, visual motion in the scene is roughly organised along a tree, with the dominant motion (typically induced by camera motion) at the root, and motion components adding up along the branches (Fig. 1). Discovering this structure would provide insight into the scene and, as a by-product, a hierarchical motion-based segmentation of it.

Object segmentation in videos is a generic problem with far-reaching applications. As such, it has received lots of attention in the computer vision literature where both fully automatic and user-assisted pipelines are proposed. Automatic segmentation tools provide key building blocks for solving problems like action localisation for instance, *e.g.*, [17]. On the other hand, interactive tools for video object segmentation are at the heart of complex video editing tasks such as cutout and rotoscoping, *e.g.*, [1, 3] and can be used to ease the arduous tasks of video annotation, *e.g.*, [31].

For tasks like tracking, segmenting, editing and analysing objects in videos, point trajectories appear as very powerful primitives: they can be harvested in large quantity and with good quality by modern techniques, *e.g.* [5], and they capture short-to-long term scene information at the fragment level. Grouping semi-dense point tracks through spatio-temporal clustering or labelling has in particular been explored in the context of object segmentation, *e.g.*, [5, 12, 13, 22, 24, 25, 29, 35]. Recent works, *i.e.* [22], report promising results on video segmentation benchmarks like the FMBS-59 dataset [25]. A number of problems remain nonetheless open, such as the definition of suitable similarities within pairs or groups of trajectories with different lifespans and the high computational complexity incurred by clustering dense tracks over long videos. More importantly for present work, existing methods lack a natural notion of compositional hierarchy.

We propose to introduce such a notion in the analysis and the clustering of point trajectories. Indeed, as observed in [15, 20], point trajectories likewise instantaneous motions result from the staked contributions of sets of dynamic parts. We found that dictionary learning and sparse coding provide appealing tools to disentangle this latent hierarchical structure. To this end, we introduce a new tree-structured dictionary learning method that allows describing each track with a few basis functions, all but one being inherited from its parent in the structure. The sparse codes thus associated to the tracks capture the desired structure and lend themselves naturally to hierarchical clustering of the collection.

The rest of the paper is organised as follows. In Section 2, we discuss relevant literature. We then introduce in Section 3 the proposed dictionary learning technique to model point trajectories with a compositional hierarchy. Section 4 is devoted to the presentation of different experiments where collections of point tracks are analysed and clustered. Finally, we give concluding remarks in Section 5.

2 Related work

In this section, we discuss the relevant literature on three key aspects of our work: the compositional hierarchical modelling of visual motion, the problem of representing and clustering point tracks found in a video sequence and, finally, the learning of dictionaries under hierarchical structure constraints.

Hierarchical motion estimation The idea of analysing visual motion through compositional hierarchies has a long history. As soon as early works on the estimation of optical flow, this concept appeared in the form of incremental coarse-to-fine estimation, either to speed-up computations in the spirit of multigrid methods [10] or to facilitate non-convex minimisation in presence of large motions [4, 26]. Combining additively dense optic flow and piece-wise parametric motions as in [21] also amounts to using a hierarchy, though a shallow one. In these works, the final goal remains however the estimation of a single motion field, whether dense or parametric, at the pixel level. Closer to our goal, several motion-based image segmentation techniques exploit nested parametric models [8, 27, 28]. In contrast to [27], and [8], where the structure is shallow and provides only a flat motion-based segmentation that captures independent moving regions in front of a background with dominant motion, [28] proposes to use deeper structures that explain locally nested motions through a hierarchical conditional random field. In any case, the above mentioned works concern the motion between two successive video frames, as opposed to the shot-level analysis that we conduct using point trajectories.

Representing and clustering point trajectories A number of recent approaches to trajectory grouping make us of a spectral embedding [6, 14, 24, 25]. Based on a suitable similarity measure between trajectories, a pairwise affinity matrix is built and used to produce a low-dimensional embedding for each trajectory (based on the bottom eigen-vectors of the associated graph Laplacian). Clustering is then conducted in this embedded space, *e.g.*, through K-means clustering in the case of spectral clustering. For this segmentation step, Brox and Malik [6] minimise instead a clustering cost that also enforces cluster separability and penalises model complexity, Ochs and Brox [24] consider higher order relationships between trajectories, and Ochs *et al.* [25] extends [6] using an MRF-based spatial prior. Our approach also proceeds through encoding of trajectories followed by clustering. However, the encoding is obtained with a special form of dictionary learning and the clustering relies on hierarchical K-means. As opposed to the methods discussed above, we obtain a hierarchical partition of the track set. Also, we outperform [25] on their FMBS-59 benchmark.

As an alternative to spectral embedding, which requires pairwise similarities, the preliminary low-dimensional encoding of the point tracks can be obtained through low-rank factorisation of the data matrix [7, 9, 29, 34, 35]. Non-negative matrix factorisation (NMF) and semi-non-negative matrix factorisation (SNMF) are for instance used in [7] and [23] respectively. Dictionary learning is also a form of data matrix factorisation, but under sparsity rather than rank constraints. To our knowledge, it has not been used so far to encoding point trajectories. Exploiting the expression power offered by a large dictionary, we propose a simple way to enforce the desired tree-based structure into the learning and the encoding steps. It is not clear whether low-rank factorisation methods are amenable to this structuring. While the SNMF-based approach of [23] for instance does capture low-level motion segments, it does not have an explicit mechanism to extract higher-level motion segments.

Once encoded through data factorisation, the trajectories can be clustered with off-the-shelf or specially designed techniques. As with spectral clustering, K-means clustering is a simple option but more sophisticated alternatives, such as multiple subspace learning, have also been investigated [9, 29]. As already said, we adopt top-down hierarchical K-means for it fits ideally our aim of unveiling the hierarchical nature of visual motion.

Structured dictionary learning and sparse coding Representing data as sparse codes over learned dictionaries is a very powerful paradigm to process or analyse collections of signals, including images and image patches within an image. Among the numerous tools that have been developed in this domain, several forms of structured sparsity have been explored in conjunction with dictionary learning, *e.g.*,[16, 18, 36]. The tree-based structured sparsity introduced by Jenatton *et al.* [18] is particularly interesting for our approach. The atoms of the dictionary being attached to the nodes of a rooted tree, this approach imposes that an input signal is encoded only with atoms that form a (small) rooted sub-tree. As we shall see in Section 3.1, our requirement is more drastic: only atoms forming a branch to the root can be jointly used to encode a given trajectory.

3 Proposed method

We start by introducing sparse coding of point trajectories. We then explain in Section 3.1 how such a learned representation can be hierarchically structured in order to capture the compositional nature of apparent motion. We extend this hierarchical sparse coding framework in Section 3.2, so that it facilitates unsupervised hierarchical clustering of the input data. Finally, we explain in Section 3.3 how trajectory clusters thus obtained over short time intervals can be regrouped at the video shot level.

In this work, we exploit point trajectories extracted with the method of Sundaram *et al.* [32], which relies on forward/backward optical flows from [6]. Although the results might be improved by using more recent optical flow methods, like the ones in [30] or [11], we stick to [6] for optical flow computation in order to perform a fair comparison with other methods, that is, only on the basis of the proposed representation and associated algorithms.

Given an input video sequence of $M+1$ frames and N input point trajectories extracted from it [1] $(\mathbf{x}^n_{0:M}) \in \mathbb{R}^{2\times(M+1)}$, $n = 1 \cdots N$, we define the data matrix $X \in \mathbb{R}^{2M\times N}$ as:

$$X = \begin{bmatrix} \Delta\mathbf{x}^1_1 & \Delta\mathbf{x}^2_1 & \cdots & \Delta\mathbf{x}^N_1 \\ \Delta\mathbf{x}^1_2 & \Delta\mathbf{x}^2_2 & \cdots & \Delta\mathbf{x}^N_2 \\ \vdots & \vdots & \cdots & \vdots \\ \Delta\mathbf{x}^1_M & \Delta\mathbf{x}^2_M & \cdots & \Delta\mathbf{x}^N_M \end{bmatrix}, \tag{1}$$

where $\Delta\mathbf{x}^n_m = \mathbf{x}^n_m - \mathbf{x}^n_{m-1}$. In this matrix, each column stems for the sequence of displacements along one trajectory. A powerful way to discover multiple structures in such data is through sparse coding with a learned dictionary. Formally, one seeks an approximate decomposition $X \approx DA$ into a *dictionary* matrix $D = [\mathbf{d}_1 \cdots \mathbf{d}_K] \in \mathbb{R}^{2M\times K}$, possibly with K larger than $2M$ (overcomplete dictionary), and a *sparse representation* $A = [\alpha_1 \cdots \alpha_N] \in \mathbb{R}^{K\times N}$ of

[1] We assume for simplicity that all trajectories are defined over the full temporal extent of the video sequence. Taking into account trajectories with different lifespans is readily done by introducing an appropriate masking matrix $P \in \{0,1\}^{2M\times N}$ in following derivations.

the input data. The columns of the matrix D are K unit-norm basis elements termed *atoms*, and those of A are the sparse codes associated to the N input trajectories. Such a sparse decomposition can be achieved by solving the optimization problem:

$$\arg\min_{D,A} \|X - DA\|_2^2 \quad \text{sb.t. } \|\alpha_n\|_0 \leq s,\ \forall n \quad \text{and} \quad \|\mathbf{d}_k\|_2 = 1,\ \forall k \tag{2}$$

using, for example, the K-SVD algorithm [2]. The positive parameter s controls the sparsity constraint and $\|X - DA\|_2$ is the reconstruction error of the trajectory displacements. At each iteration of the K-SVD algorithm, a coding step is performed (*i.e.*, solving for A in Eq. 2, dictionary D being fixed) with the orthogonal matching pursuit (OMP)[33], followed by an SVD-based update of the dictionary's atoms. The previous formulation, however, does not enforce any structure among the atoms of the dictionary and on the associated codes. Next, we re-formulate the problem so that the dictionary and the encoding are constrained in certain way by a tree structure.

3.1 Tree-structured dictionary learning

As pointed out in [15], the natural organization of the moving objects and their parts in a video is that of a tree, as illustrated in Fig. 1. In this simple example, the motion of one leg of the bear constitutes one node of the tree with the motion of the animal's body being at its parent node, the latter being in turn related to the root node that captures the visual motion induced in the whole scene by the movement of the camera. In fact, this hierarchical organization of visual motion is not restricted to articulated objects, but to other natural scenes as well. We aim at leveraging such a structure.

A second key feature of our approach is to represent point tracks as linear combinations of few learned atoms. We thus resort to dictionary learning and sparse coding techniques, with the goal of organizing the dictionary in a hierarchical structure that can capture, to some extent, the compositional organization of the dynamic scene. Each trajectory pattern in the dictionary is associated to a node of a tree and should ideally capture the motion of one scene element, *relative to its ancestors in the tree*. In other words, we want the movement of a given scene element to be represented *only with dictionary atoms stemming from a same branch of the tree*. This form of hierarchical tree-structured sparsity is related to the one of Jenatton *et al.* [18], but is more drastic. Jenatton *et al.* indeed stipulate that only a sub-tree from the root can be used to represent an input signal.

For a given rooted tree $\mathcal{T}$ of K nodes numbered in level-order,[2] we want to learn a dictionary $D = [\mathbf{d}_{1:K}] \in \mathbb{R}^{2M \times K}$ of K trajectory atoms organized according to this tree structure, together with the corresponding matrix $A = [\alpha_{1:N}] \in \mathbb{R}^{K \times N}$ of sparse codes. To this end, we consider the following constrained minimization problem:

$$\arg\min_{D,A} \|X - DA\|_2^2, \quad \text{sb.t. } \alpha_n \in \mathcal{A}(\mathcal{T}),\ \forall n \quad \text{and} \quad \|\mathbf{d}_k\|_2 = 1,\ \forall k, \tag{3}$$

where $\mathcal{A}(\mathcal{T}) \subset \mathbb{R}^K$ is the set of tree-structured codes defined as:

$$\mathcal{A}(\mathcal{T}) = \{\alpha \in \mathbb{R}^K : \text{supp}(\alpha) = \text{anc}(\text{k}(\alpha))\}, \tag{4}$$

where $\text{anc}(k)$ denotes the ancestor set of node k in $\mathcal{T}$ (the nodes, including itself, that form the unique path from k to root node 1), $\text{supp}(\alpha)$ is the support of α, that is the index set

[2] In practice, a simple regular structure is chosen, defined by its depth L, and by the common number n_ℓ, $\ell = 1 \cdots L-1$, of children for all nodes at a given depth level ℓ. In that case, $K = 1 + \sum_{\ell=1}^{L-1} \prod_{\ell'=1}^{\ell} n_{\ell'}$.

of its non-zero entries, and $\mathrm{k}(\alpha) = \max(\mathrm{supp}(\alpha))$ stands for the last atom in the code. In other words, only a single branch of the tree, from the root to a certain node which is not necessarily a leaf, can be used to encode a given point track. This constraint also enforces the sparsity of the codes since $\|\alpha\|_0$ cannot exceed the depth of the tree.

Algorithm 1 Tree-structured orthogonal matching pursuit

1: **procedure** TREE-OMP(`dictionary` D, `signal` $\mathbf{x}$, `target error` ε, `tree` $\mathcal{T}$)
2: $\quad\mathcal{S} \leftarrow \{1\}$
3: $\quad\alpha_{\mathcal{S}} \leftarrow \mathbf{d}_1^\top \mathbf{x}$
4: $\quad\mathbf{r} \leftarrow \mathbf{x} - \mathbf{d}_1 \alpha_{\mathcal{S}}$
5: $\quad$**while** $\|\mathbf{r}\|_2 > \varepsilon$ and $\mathrm{size}(\mathcal{S}) < \mathrm{depth}(\mathcal{T})$ **do**
6: $\quad\quad k \leftarrow \arg\max_{k \in \mathrm{child}(\max \mathcal{S})} |\mathbf{d}_k^\top \mathbf{r}|$
7: $\quad\quad\mathcal{S} \leftarrow \mathcal{S} \cup \{k\}$
8: $\quad\quad\alpha_{\mathcal{S}} \leftarrow D_{\mathcal{S}}^{+} \mathbf{x}$
9: $\quad\quad\mathbf{r} \leftarrow \mathbf{x} - D_{\mathcal{S}} \alpha_{\mathcal{S}}$
10: $\quad$**return** (α)

We use the K-SVD algorithm to solve the dictionary learning problem (3), but we modify the orthogonal matching pursuit (OMP) encoding part in order to respect the constraint in (4), as it can be observed in Algorithm 1. We force the use of the root node for every input datum, so in line 2, the code support $\mathcal{S}$ is initialized with $\{1\}$. In subsequent steps, the greedy search for a new atom to include, in line 6, is restricted to the children of the last selected atom in the tree. In line 8, $D_{\mathcal{S}} = [\mathbf{d}_k]_{k \in \mathcal{S}}$ is the sub-dictionary indexed by $\mathcal{S}$ and $\alpha_{\mathcal{S}}$ is the corresponding code for $\mathbf{x}$, obtained through least-squares minimization, with $(.)^+$ standing for matrix pseudo-inverse. The algorithm stops when the reconstruction is accurate enough or the maximum tree depth is reached.

3.2 Hierarchical coding and clustering

Having all trajectories encoded in A according to the learned tree-structured dictionary D already provides a flat partitioning of the trajectories through indices $\mathrm{k}(\alpha_n)$ and a hierarchical one by gathering, for each node k in the tree, all trajectories such that $k \in \mathrm{anc}(\mathrm{k}(\alpha))$. Unfortunately, such partitions are noisy since the above dictionary learning and sparse coding are not explicitly geared toward a clustering task: nothing prevent unrelated trajectories to share atoms and, conversely, related trajectories to exhibit disjoint supports.

In the same spirit as spectral clustering that conducts final K-means clustering on spectrally encoded data vectors instead of simply binarising them, we can cluster the trajectories based on their codes α_ns, with hierarchical K-means in our case. This already provides cleaner partitions. Drawing inspiration from Jiang *et al.* [19] who combine dictionary learning with supervised learning of linear classifiers over codes, we can go one step further: given a current hierarchical clustering of trajectories' codes, we can update our tree-structured dictionary and iterate. As will appear in the experiments, this procedure further improves the quality of track clusters. At each iteration, the dictionary learning problem to solve becomes

$$\arg\min_{D,Y,A} \|X - DA\|_2^2 + \lambda \|Q - YA\|_2^2, \quad \text{sb.t. } \alpha_n \in \mathcal{A}(\mathcal{T}),\ \forall n \tag{5}$$

where $Q \in \{0,1\}^{K \times N}$ is the binary matrix associated to the current hierarchical clustering of

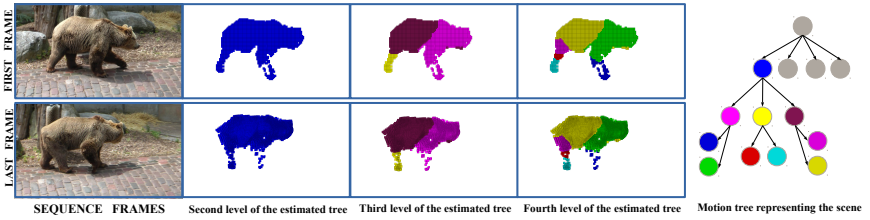

Figure 2: Hierarchical motion segmentation of the *bear* sequence from FMBS dataset [25]. In this sequence, the camera is following a walking bear. The moving parts discovered at each level of the tree are showed in first and last frames, along with associated colour-coded tree. The segments associated to the greyed nodes are not visualized for sake of readability (those from the second level of the tree are assigned to background points).

tracks (each of its columns belongs to $\mathcal{A}(\mathcal{T})$) and λ is a positive parameter that controls the balance between reconstruction and clustering terms. We set λ by trial and error, and fix its value to 1 for all the experiments. This new objective function can be rewritten as

$$\|X - DA\|_2^2 + \lambda \|Q - YA\|_2^2 = \Big\| \underbrace{\begin{bmatrix} X \\ \sqrt{\lambda} Q \end{bmatrix}}_{\bar{X}} - \underbrace{\begin{bmatrix} D \\ \sqrt{\lambda} Y \end{bmatrix}}_{\bar{D}} A \Big\|_2^2, \tag{6}$$

and optimized w.r.t. $\bar{D}$ and A with K-SVD, after trading normalization constraints on D and Y for normalization constraints on $\bar{D}$.

3.3 Extension to longer videos

With the proposed approach, we are able to extract meaningful part-based clusters of tracks from short video sequences, *i.e.*, 20 frames at most. Even over such short time intervals, some of the trajectories are incomplete. We handle them by computing the reconstruction error for a trajectory only in frames it is defined, *i.e.*, replacing $\|X - DA\|_2$ by $\|P \odot X - P \odot DA\|_2$, where P is the binary masking matrix that sets to zero undefined entries of X and $\odot$ is the Hadamard product. An example of hierarchical track clustering obtained over a short sequence is shown in Figure 2.

To process longer video shots of, say, hundreds of frames, we apply our method independently to half-overlapping chunks of 10 frames each and we follow the spectral clustering method in [23] to group short-term clusters at the shot level. Given all finest-level segments (groups of tracks associated to tree leaves) extracted from all temporal windows through tree-structured spare coding, we build a pairwise affinity matrix based on the number of tracks that are shared by each pair of segments. This affinity matrix is used to conduct spectral clustering. As a result, two groups of trajectories that have been formed with our approach in two distinct time windows are likely to be merged in the final segmentation if they share a lot of tracks. This might happen even if they are quite distant in time, provided long-term trajectories exist in the initial data and some of them belong to both groups.

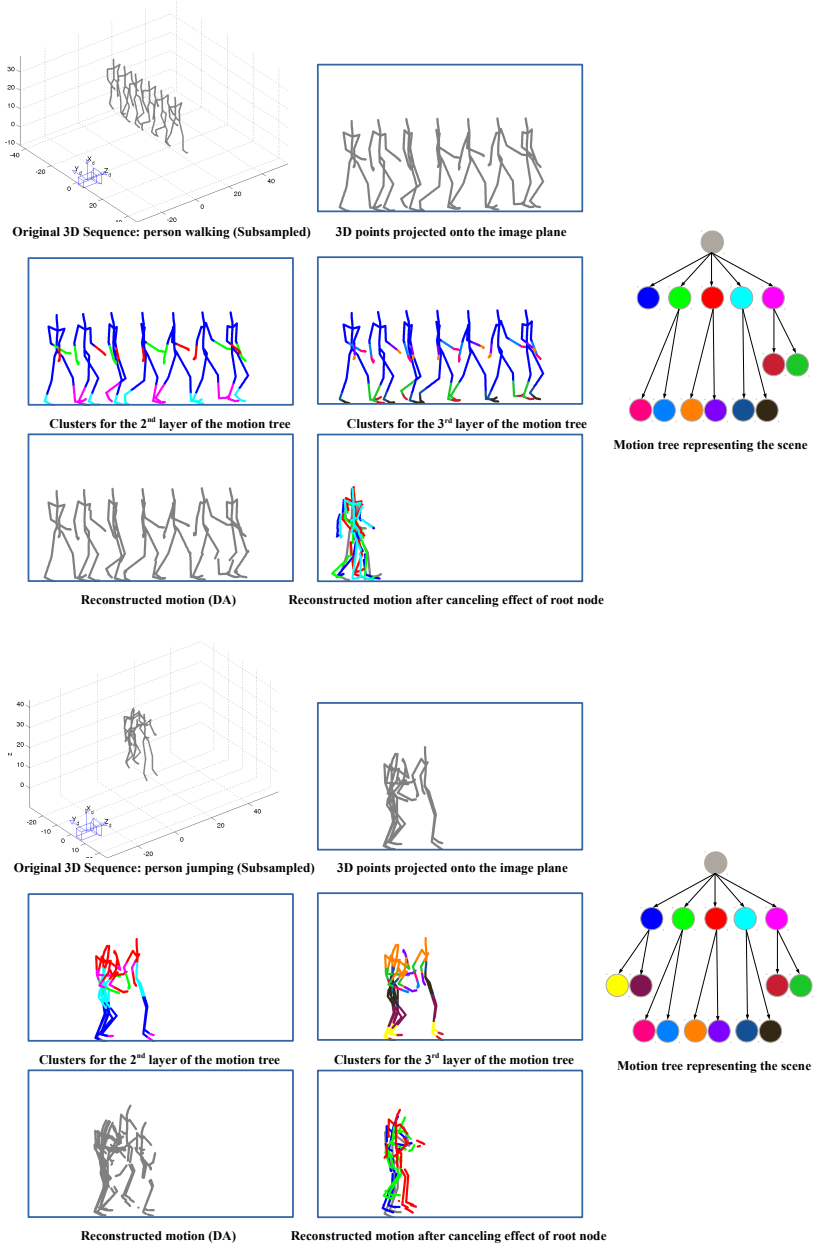

Figure 3: Hierarchical motion analysis of *walking* and *jumping* sequences from CMU Mo-Cap dataset. For each sequence: (Top) Original 3D sequence and its 2D projection in the virtual camera 2D; (Middle) Clustering results on the 2nd and 3rd levels of the tree and corresponding colour-coded tree. (Bottom) Motion reconstructed from complete codes and after removing the contribution of root track atom $k = 1$.

Table 1: Object segmentation results on the FMBS-59 test set.

	Average precision	Average recall	Average F-measure
Spectral Clustering [25]	76.15%	61.11%	67.81%
Multicuts [22]	81.04%	68.67%	74.34%
DL (baseline)	67.84%	58.81%	60.25%
TreeDL (ours)	76.84%	64.20%	69.46%
TreeDL$^+$ (ours)	78.41%	65.52%	72.33%

4 Experiments

Using the CMU MoCap dataset,[3] we analyse first the ability of our approach to discover motion hierarchies. From these real 3D human motion data, we can derive a structured set of 2D point trajectories: around 1500 points are sampled from the moving limbs and projected into an arbitrary virtual camera, where they produce 2D tracks. We aim at discovering a plausible hierarchical decomposition of this data, *i.e.*, one that complies to some extent with the kinematic chain of the articulated human body. For these experiments, we use a simple tree structure $\mathcal{T}$ composed of 1 root node with five children, each one with two children ($L = 3,\ n_1 = 5,\ n_2 = 2,\ K = 16$). Note that some nodes might be unused in the end, no trajectories being assigned to them. It is the case in the *walking* sequence in Fig. 3, for two siblings of the last level of the tree. The quality of the obtained sparse approximation $X \approx AD$ can be assessed through visualization of the corresponding track reconstructions (Fig. 3, third row left). Also, in order to get insight into what a specific track atom k captures, we can simply set to zero the corresponding k-th row in the code matrix A and recompute the reconstructed trajectories accordingly. We show in particular the effect of removing the influence of the root node in *walking* and *jumping* sequences (Fig. 3, third row right). In both cases, the root atom has captured the global trajectory of the actor. The reconstructed motion depicts the actor performing approximately the same actions, but in place. Note that each trajectory is reconstructed from at most three atoms (number of levels in the tree), which can lead to some reconstruction errors. However, despite the simplicity of the underlying tree, our approach is able to discover automatically meaningful structures among the input trajectories. In the *jumping* case, the left and right parts of the body are moving in the same way, this explains why trajectories from limb pairs, *e.g.* the two feet, are grouped together.

In a second series of experiments, we evaluate the clustering performance of our motion analysis framework on the FMBS-59 dataset [25].We present in Table 1 the results of our tree-structured dictionary learning approach with iterative refinement ("TreeDL$^+$") as described in Section 3.2 and compare them against state-of-the-art methods for trajectory clustering, namely the spectral clustering based method in [25] and the multicuts-based approach in [22]. For our approach, the tree structure is defined by $L = 4,\ n_1 = 4,\ n_2 = 3,\ n_3 = 2$ and $K = 41$. We also include results for two baselines related to our approach: using unstructured dictionary learning ("DL") as introduced in Section 3 and a hierarchical learned representation with no further refinement ("TreeDL"), as explained in Section 3.1. For these two baselines and our complete system, we use the temporal sliding-window procedure described in Section 3.3 and use top-down hierarchical K-means on final codes to produce track clusters in each video chunk. We first note in Table 1 the gain brought by the tree-based structuring and by the clustering-driven extension of dictionary learning. We also note that both "TreeDL" and "TreeDL$^+$" outperform [25]. While the performance of proposed system is

[3] http://mocap.cs.cmu.edu/

slightly below the one of [22], our system gives access to precious information about the structure of the motion and the way the moving regions relate to each other.

5 Conclusions

We propose a method for representing and clustering point tracks, which captures the natural organization of moving regions in a dynamic scene. Our approach relies on an original dictionary learning technique that enforces a tree-based structure of dictionary and codes, and takes explicitly into account the subsequent task of clustering final codes. We showed experimentally that our method not only performs well on the difficult task of trajectory-based video segmentation, but also discovers automatically part of the hierarchical structure of dynamic scenes, both in motion capture data and in monocular video data.

References

[1] Aseem Agarwala, Aaron Hertzmann, David H Salesin, and Steven M Seitz. Keyframe-based tracking for rotoscoping and animation. *ACM Transactions on Graphics (ToG)*, 23(3):584–591, 2004.

[2] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

[3] Xue Bai, Jue Wang, David Simons, and Guillermo Sapiro. Video snapcut: robust video object cutout using localized classifiers. *ACM Transactions on Graphics (ToG)*, 28(3): 70, 2009.

[4] Michael J Black and Paul Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer vision and image understanding*, 63(1):75–104, 1996.

[5] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *ECCV, Heraklion*, 2010.

[6] Tomas Brox, Christoph Bregler, and Jitendra Malik. Large displacement optical flow. In *CVPR, Miami*, 2009.

[7] Anil M Cheriyadat and Richard J Radke. Non-negative matrix factorization of partial track data for motion segmentation. In *CVPR, Miami*, 2009.

[8] Gabriella Csurka and Patrick Bouthemy. Direct identification of moving objects and background from 2D motion models. In *ICCV, Kerkira*, 1999.

[9] Ehsan Elhamifar and René Vidal. Sparse subspace clustering. In *CVPR, Miami*, 2009.

[10] Wilfried Enkelmann. Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences. *Computer Vision, Graphics, and Image Processing*, 43(2):150–177, 1988.

[11] Denis Fortun, Patrick Bouthemy, and Charles Kervrann. Aggregation of local parametric candidates with exemplar-based occlusion handling for optical flow. *Computer Vision and Image Understanding*, 145:1–182, 2015.

[12] Matthieu Fradet, Philippe Robert, and Patrick Pérez. Clustering point trajectories with various life-spans. In *CVMP, London*, 2009.

[13] Katerina Fragkiadaki, Weiyu Zhang, Geng Zhang, and Jianbo Shi. Two-granularity tracking: Mediating trajectory and detection graphs for tracking under occlusions. In *ECCV, Florence*, 2012.

[14] Katerina Fragkiadaki, Pablo Arbelaez, Panna Felsen, and Jitendra Malik. Learning to segment moving objects in videos. In *CVPR, Boston*, 2015.

[15] Samuel J Gershman, Joshua B Tenenbaum, and Frank Jäkel. Discovering hierarchical motion structure. *Vision Research*, 2015. In press.

[16] Junzhou Huang, Tong Zhang, and Dimitris Metaxas. Learning with structured sparsity. *The Journal of Machine Learning Research*, 12:3371–3412, 2011.

[17] Mihir Jain, Jan Van Gemert, Hervé Jégou, Patrick Bouthemy, and Cees GM Snoek. Action localization with tubelets from motion. In *CVPR, Columbus*, 2014.

[18] Rodolphe Jenatton, Julien Mairal, Guillaume Obozinki, and Francis Bach. Proximal methods for sparse hierarchical dictionary learning. In *ICML, Haifa*, 2010.

[19] Zhuolin Jiang, Zhe Lin, and Larry S Davis. Learning a discriminative dictionary for sparse coding via label consistent K-SVD. In *CVPR, Colorado Springs*, 2011.

[20] Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Attention, Perception, & Psychophysics*, 14(2):201–211, 1973.

[21] Shanon X Ju, Michael J Black, and Allan D Jepson. Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In *CVPR, San Francisco*, 1996.

[22] Margret Keuper, Bjoern Andres, and Thomas Brox. Motion trajectory segmentation via minimum cost multicuts. In *ICCV, Santiago*, 2015.

[23] Quanyi Mo and Bruce A Draper. Semi-nonnegative matrix factorization for motion segmentation with missing data. In *ECCV, Florence*, 2012.

[24] Peter Ochs and Thomas Brox. Higher order motion models and spectral clustering. In *CVPR, Providence*, 2012.

[25] Peter Ochs, Jagannath Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1187–1200, 2014.

[26] Jean-Marc Odobez and Patrick Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6 (4):348–365, 1995.

[27] Jean-Marc Odobez and Patrick Bouthemy. Direct incremental model-based image motion segmentation for video analysis. *Signal Processing*, 66(2):143–155, 1998.

[28] Juan-Manuel Pérez-Rúa, Tomas Crivelli, Patrick Pérez, and Patrick Bouthemy. Hierarchical motion decomposition for dynamic scene parsing. In *ICIP, Phoenix*, 2016.

[29] Shankar R Rao, Roberto Tron, René Vidal, and Yi Ma. Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In *CVPR, Anchorage*, 2008.

[30] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. EpicFlow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR, Boston*, 2015.

[31] Naveen Shankar Nagaraja, Frank R Schmidt, and Thomas Brox. Video segmentation with just a few strokes. In *ICCV, Santiago*, 2015.

[32] Narayann Sundaram, Tomas Brox, and Kurt Keutzer. Dense point trajectories by GPU-accelerated large displacement optical flow. In *ECCV, Hersonissos*, 2010.

[33] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007.

[34] René Vidal and Richard Hartley. Motion segmentation with missing data using power-factorization and GPCA. In *CVPR, Washington DC*, 2004.

[35] René Vidal and Yi Ma. A unified algebraic approach to 2-D and 3-D motion segmentation. In *ECCV, Prague*, 2004.

[36] Peng Zhao, Guilherme Rocha, and Bin Yu. The composite absolute penalties family for grouped and hierarchical variable selection. *The Annals of Statistics*, pages 3468–3497, 2009.